Energy-Efficient Service Chains with Network Function Virtualization

Andrea Tomassilli, Nicolas Huin, Frédéric Giroire, Brigitte Jaumard

Energy-Efficient Service Chains with Network Function Virtualization

Andrea Tomassilli, Nicolas Huin, Frédéric Giroire, and Brigitte Jaumard

RESEARCH REPORT
N° 8979
October 2016
Project-Team Coati

ISSN 0249-6399 ISRN INRIA/RR--8979--FR+ENG

Energy-Efficient Service Chains with Network Function Virtualization

Andrea Tomassilli*, Nicolas Huin*, Frédéric Giroire*, and Brigitte Jaumard†

Project-Team Coati

Research Report n° 8979 — October 2016 — 14 pages

Abstract: Service Function Chains (SFCs) are an ordered sequence of network functions, such as firewall. Using the new approaches of Software Defined Networks and of Network Function Virtualization (NFV), the network functions can be virtualized and executed on generic hardware. To optimize network management, it is thus crucial to place dynamically the network functions at the right positions in the network according to the network traffic. In this paper, we consider the problem of SFC placement with the goal of minimizing network energy consumption. We model the problem as an Integer Linear Program, which can be used to solve small instances. To solve larger instances, we propose GreenChains, a heuristic algorithm. We exhibit the benefit of dynamic routing and of NFV on the energy savings. We show that between 30 to 55% of energy can be saved for typical ISP networks, while respecting the SFC constraints.

Key-words: Network Function Virtualization, Service Function Chains, Software Defined Networks, Energy Efficiency, Optimization

* A. Tomassilli, N. Huin, and F. Giroire have been partially supported by ANR program ANR-11-LABX-0031-01 and Mitacs.

† B. Jaumard was supported by a Concordia University Research Chair (Tier I) and by an NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Chaîne de services efficaces en énergie grâce à la virtualisation des fonctions réseaux

Résumé : Les chaînes de fonctions de service (SFC) sont une séquence ordonnée de fonctions réseaux. En utilisant les nouvelles approches des réseaux logiciels (Software Defined Networks en anglais) et de virtualisation des fonctions réseaux (NFV), les fonctions réseaux peuvent être virtualisée et exécutée sur des équipements génériques. Pour optimiser la gestion des réseaux, il est crucial de placer de façon dynamique les fonctions réseaux aux positions adéquates dans le réseau en fonction du traffic. Dans ce papier, nous considérons le problème de placement de SFC avec l'objectif de minimiser la consommation énergétique des réseaux. Nous modélisons le problème par un programme linéaire en nombres entiers qui peut résoudre de petites instances. Pour des instances plus grandes, nous proposons GreenChains, un algorithme heuristique. Nous exhibons les bénéfices du routage dynamique et de NFV sur les gains en énergie. Nous montrons que de 30 à 50% d'énergie peut être sauvée pour des réseaux typiques d'opérateurs tout en respectant les contraintes des SFC.

Mots-clés : Virtualisation des fonctions réseaux, Chaînes de fonctions de service, Réseaux logiciels, Efficacité énergétique, Optimisation

1 Introduction

With the large yearly increase of Internet traffic, future networks will have to be more energy efficient [1]. One of the classic methods to reduce the energy consumption of networks is to try to aggregate network traffic on a small number of network equipments in order to put to sleep the unused hardware. However, a difficulty is that today's traffic has to pass through a number of network functions, such as deep packet inspection (DPI), firewall, load balancing, WAN optimization. The network functions often need to be applied in a specific order, e.g. in a security scenario, the firewall has to be applied before carrying out a DPI as the last one is more CPU intensive. In this context, a *Service Function Chain (SFC)* is a list of network functions, that need to be applied to a flow in a particular order. These functions are carried out by specific hardware, which are installed at specific locations of the network. The paths followed by demands are thus very constrained, reducing the opportunities to aggregate traffic.

With the emergence of technics of Network Function Virtualization (NFV), the functions can now be executed by generic routers instead of dedicated equipments. Coupled with the Software Defined Network (SDN) paradigm, NFV brings a great flexibility to manage network flows. Indeed, with the centralized control allowed by SDN, the flow can be managed dynamically from end-to-end and the service functions can be installed only along paths for which and when they are necessary. These new paradigms thus bear the opportunity for energy savings in networks.

In this work, we explore the potential energy savings of using NFV for Service Function Chains. We consider the problem of reducing network energy consumption while placing service functions using generic hardware along the paths followed by flows. A specific difficulty is that the network functions can be repeated several time in the same chain and have to be executed in a specific order.

In summary, the contributions of this work are

- We show how virtualization can be used to improve the energy efficiency of networks, when demands have to go through a chain of services. To the best of our knowledge, we are the first to propose such a method.

- We propose a new way of modeling this problem based on Integer Linear Programming (ILP). The ILP can solve optimally instances of small sizes. We thus propose and validate a *heuristic algorithm*, GreenChains, to handle instances of larger sizes.

- This allows us to carry out extensive simulations on networks of different sizes. We study three different scenarios: a *legacy scenario* which serves as baseline for comparison, a *hardware scenario* in which the routing can be changed dynamically by a centralized SDN controller, but in which network functions are executed by specific hardware, and finally, an *NFV scenario* in which the network functions are virtualized and can be placed dynamically. We show that between 30 to 55% of energy can be saved while respecting the constraints of the service chains.

- We additionally study the impact of the number of occurences of Virtual Network Functions (VNF) on bandwdith usage, path delay, and energy savings. We show that with only a small number of replicates near optimal results can be obtained.

The remainder of the paper is organized as follow. Related work is discussed in Section 1.1. The optimization problem and the ILP model are explained in Section 2, while the heuristic algorithm is described in Section 3. Numerical results are given in Section 4.

1.1 Related Work

Service Chains. Several works consider the problem of service function chain placement, but considering other metrics or other scenarios. [2] proposes a different ILP model to solve problem studies the impact of the positions of chains on the processing costs. The authors of [3] explores the joint placement and routing of traffic in order to minimize the newtork bandwidth consumption. In [4] is proposed a layered ILP close to the one we propose in the paper, but they optimize the latency. [5] explores a problem of joint optimization of maximum link, CPU core and maximum delay in the network while placing the NVFs. Last, [6] considers a cloud environment in which the load has to be load-balanced in order to minimize the computation and the communication overheads . However, these works do not consider the problem minimizing network energy consumption with a dynamic traffic.

Energy-Aware Routing. Several works have proposed algorithms to obtain energy aware routing, see e.g. [7, 8]. However, these works are hard to be put in practice as operators of legacy networks are reluctant to change their network configurations.

SDN and Network Energy Efficiency. Recently, researchers have started to explore how the introduction of the SDN paradigm with a centralized control and a live report of metrology data may allow to carry out dynamic routing. In particular, it would allow to implement energy-aware routing algorithms, as discussed in [9]. However, these papers did not consider the constraints of network functions. Some particular works considered some specific class of network functions, like compression [10], but not the general problem of ensuring that flows are treated by the network functions.

Network Virtualization and Network Energy Efficiency. Only two papers explore the potential of network virtualization for energy efficiency In [11], the author present an extension of an open source software framework, the Distributed Router Open Platform (DROP), to enable a novel distributed paradigm for NFV. DROP includes sophisticated power management mechanisms, which are exposed by means of the Green Abstraction Layer. In [12], authors estimate the energy savings that could result from the three main NFV use cases-Virtualized Evolved Packet Core, Virtualized Customer Premises Equipment and Virtualized Radio Access Network. However, both papers do not consider the constraints of service chains.

2 Model

2.1 Problem

We model our network as a bidirectional graph $G = (V, A)$. A router is represented by a node. Each node $u \in U$ has a capacity C_u to handle network functions and which can represent its CPU, memory or storage (for simplicity, we consider here that the capacity of u is given by its number of cores). A link between two routers u and v consists in two directed arcs uv and vu in opposite directions. An arc uv has a capacity C_{uv}.

The network operator has a set of demands $\mathcal{D}$ which need to be routed. A demand is a 4-tuple $(s, t, D^{st}, \gamma^{st})$, where s is the source of the demand, t its destination, and D^{st} the amount of flow which has to be routed from s to t. The demand has to pass through a chain of network functions among $\mathcal{F}$, the set of the network NFVs. We note the chain $\gamma^{st} = f_1^{st}...f_{L^{st}}^{st}$, with L^{st} the chain length. A function f_i can be more or less demanding for a node (e.g. it is more resource intensive to compress traffic, than to simply load balance it). To model it, we note Δ_f the fraction of cores used by function f per bandwidth unit. For different reasons (e.g. resiliency, privacy), two different functions f_i and f_j may not be allowed to be installed in the same node. To model this, we define θ_i, the set of anti-affinity functions that cannot be installed at the same location as f_i.

*The Energy Efficient Service Function Chain Placement Problem (*EE-SFCP*) problem consists in routing the set of demands $\mathcal{D}$ to minimize the network energy consumption, while respecting link and node capacities and constraints given by the set of chains γ_{st}.*

Power Model. Campaigns of measures of power consumption (see e.g. [13]) show that a network device consumes a large amount of its power as soon as it is switched on and that the energy consumption does not depend much of the load. Following this observation, On/Off power models have been proposed and studied. Later, researchers and hardware constructors have proposed more energy proportional hardware [14]. To cover the different models for different hardware, we use a hybrid power model. Power of an active link:

$$\text{POW}_{uv} + \frac{\text{FLOW}_{uv}}{C_{uv}} \text{POW}'_{uv}$$

In our model, we considered that links can be put into sleep mode (by putting to sleep both corresponding interfaces). Note that the two links (uv and vu) between a pair of nodes are in the same state (active or in sleep mode) as there are sharing the same interface. But the routers themselves may not be put into sleep mode, as there are the sources or destinations of network traffic. However, cores may be put into sleep mode and the power used by nodes is given by

$$\text{POW}_u \times \#\text{cores}$$

In the experiments, we consider that $\text{POW}_{uv} = \text{POW}'_{uv} = \text{POW}_u$.

2.2 A layered Model

We model the problem as an Integer Linear Program (ILP) using a layered model to express the orders between functions of the service chains. The model has $L^{st}+1$ layers. We note $u(i)$ the copy of Node u in Layer i. The paths for the demand D^{st} starts from node $s(0)$ of layer 0 and ends in node $t(L^{st})$ of layer L^{st}. Passing from layer i to layer $i+1$ means that the i^{th} function of the chain has been applied.

2.3 Notations

$G = (V, A)$	
$\mathcal{F}$	set of functions
$\mathcal{D}$	set of Demands
C_{uv}	capacity of link (u, v)
$\gamma^{st} = f_1^{st} ... f_{L^{st}}^{st}$	chain for demand between s and t, with L^{st} the chain length
θ_i	set of anti-affinity functions for f_i
f_l^{st}	last function in γ_{st}
D^{st}	bandwidth demand from s to t
Δ_f	fraction of cores used by function f per bandwdith unit
C_u	capacity of node u (in # cores)

2.4 Formulation

Variables:

- $x_{uv} \in \{0,1\}$ where $x_{uv} = 1$ if link uv is active, 0 otherwise
- $x_{f_i u} \in \{0,1\}$ where $x_{f_i u} = 1$ of function f_i is installed on node u, 0 otherwise
- $g^{st}_{u(i)v(i)} \in \{0,1\}$ where $g^{st}_{u(i)v(i)} = 1$ if the flow for the demand st uses the link $u(i)v(i)$, i.e., the link uv in layer i. We consider here unsplittable routing.
- $k_u \in \mathbb{N}$, #cores used in node u.
- $f_{uv} \in \mathbb{R}$, flow passing through link uv. This variable is linked and is added to the ILP for clarity of the presentation.

Objective

$$\min \sum_{(u,v)\in A} (\text{POW}_{uv} x_{uv} + \text{POW}'_{uv} \frac{f_{uv}}{C_{uv}}) + \sum_{u \in V} \text{POW}_u k_u \tag{1}$$

Flow Conservation

$$\sum_{v \in N^+(u)} g^{st}_{u(i)v(i)} - \sum_{v \in N^-(u)} g^{st}_{v(i)u(i)} + g^{st}_{u(i)u(i+1)} - g^{st}_{u(i-1)u(i)}$$
$$= \begin{cases} 1 & \text{if } u(i) = s(0), \\ -1 & \text{if } u(i) = t(L^{st}), \\ 0 & \text{else} \end{cases} \quad \begin{array}{l} (s,t) \in \mathcal{D}, u \in V, \\ 0 \le i \le L^{st}, f_i \in \mathcal{F} \end{array} \tag{2}$$

Link Capacity

$$f_{uv} = \sum_{(s,t)\in\mathcal{D}} \sum_{i=0}^{L^{st}} D^{st} g^{st}_{u(i)v(i)} \quad (u,v) \in A$$

$$f_{uv} \le x_{uv} C_{uv} \quad (u,v) \in A \tag{3}$$

Function Activation

$$x_{f^{st}_i u} \ge g^{st}_{u(i-1)u(i)}, \quad u \in V, (s,t) \in \mathcal{D}, 1 \le i \le L^{st} \tag{4}$$

Anti-affinity rules

$$x_{f_i u} \le 1 - x_{f_j u} \qquad f_i \in \mathcal{F}, f_j \in \theta_i, u \in V \tag{5}$$

Number of cores used

$$\sum_{(s,t)\in\mathcal{D}} \sum_{i=1}^{l^{st}} \Delta_{f^{st}_i} D^{st} g^{st}_{u(i-1)u(i)} \le k_u \quad u \in V \tag{6}$$

Node Capacity

$$k_u \le C_u \quad u \in V \tag{7}$$

For Equations (2), we set $g^{st}_{u(-1)u(0)} = 0$ and $g^{st}_{u(l^{st})u(l^{st}+1)} = 0$. It allows to write the same set of equations for all the layers of the model. Note the link capacity constraints (3). If the link uv is into sleep mode, the sum of the flows forwarded through it must be null.

3 Solving large Instances with GreenChains

As the ILP proposed in the previous section cannot provide solutions for large networks, we propose here a heuristic algorithm called GreenChains to solve the EE-SFCP problem. The problem can be decomposed into three-subproblems.

- First, the *routing problem* is to compute a path for each demand, respecting the link capacity constraints.
- Second, the goal of the *service chain placement problem* is to find a placement of the NVF respecting the capacities of the nodes and the order defined by the service chains.
- Last, the *energy saving problem* tries to put into sleep mode as many links and cores as possible to decrease the energy consumption of the network.

Routing Module. The routing module is simple. The demands are considered one by one. For each demand, we compute a weighted shortest path on a residual graph. The weight of a link is equal to the inverse of its residual capacity. This favors the link with lower load. The idea is to spread as much as possible the demands to find a feasible routing. To compute the residual graph, when the path is computed, we reduce the link capacities by the value of the demand. Furthermore, when considering a new demand to be routed, we remove links with a residual capacity smaller than the demand.

Service Chain Placement Module. When the paths for all demands are set, we place the service chains. We consider the different service chains separately. For a given chain, we consider the paths of all the demands with this chain. We then sort the nodes of the networks in the decreasing order of the number of such paths they are involved in. We then try to place all the functions of the chains on the nodes following this order. If it is possible in a node u, we place the functions. We then assign some of the paths to the cores of node u, till its capacity is full. We then actualize the ordered list of nodes considering only the paths not yet assigned.

Energy Saving Module. The module tries to greedly put links into sleep mode. It starts with the graph $G = (V, E)$. It first launches the routing module and then the service chain placement module. If they both succeed, it creates a list U of all links according to their usage (volume of traffic). It then chooses the less loaded link $e_{\min}$ as a candidate to be put in sleep mode. It now considers the graph $G' = (V, E \setminus e_{\min})$. It launches again the routing and placement modules. If they succeed, $e_{\min}$ is definitely put in sleep mode. The list U is actualized with the new routing, as well as the less loaded link. If at least one of the two modules fails, GreenChains considers that $e_{\min}$ cannot be into sleep mode and the link is definitely kept active for the final solution. The second element of U is then considered. The algorithm goes on till all links have been tried and set either as definitely in sleep mode or definitely active.

4 Results

In this section, we study the EE-SFCP for three networks with different sizes of the SNDLib [15] library:

- `pdh` with 11 nodes and 64 directed links, 440 demands.
- `atlanta` with 15 nodes and 44 directed links, 840 demands.
- `germany` with 50 nodes and 176 directed links, 9800 demands.

Service Chain	Chained VNFs	rate	% traffic
Web Service	NAT-FW-TM-WOC-IDPS	100 kbps	18.2%
VoIP	NAT-FW-TM-FW-NAT	64 kbps	11.8%
Video Streaming	NAT-FW-TM-VOC-IDPS	4 Mbps	69.9%
Online Gaming	NAT-FW-VOC-WOC-IDPS	50 kbps	0.1%

Table 1: Service Chain Requirements [2]

NAT: Network Address Translator, FW: Firewall, TM: Traffic Monitor, WOC: WAN Optimization Controller, IDPS: Intrusion Detection Prevention System, VOC: Video Optimization Controller

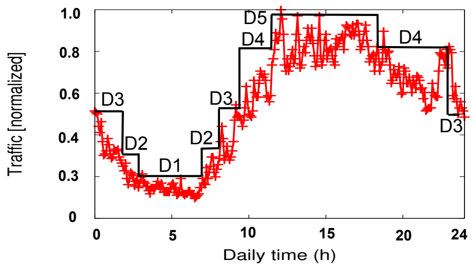


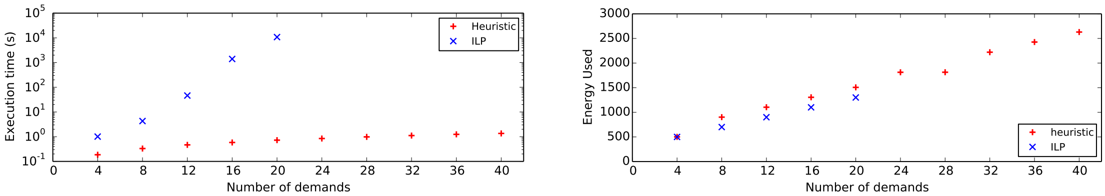
Figure 1: Multi-period approximation of the trace of the typical daily traffic of a link of France Telecom network.

We first validate the heuristic algorithm, GreenChains, by comparing its results to the optimal one found by the ILP proposed in Section 2. The ILP model can only solve small instances, we thus use GreenChains to solve larger instances. We then study the energy saved during the day for the three networks. The core of the experiments is to compare three scenarios presented below: a *legacy scenario*, a *hardware scenario* and an *NFV scenario*. We show that GreenChains succeeds in savings between 35% and 55% of the energy used in the *legacy scenario*.

Network Traffic Model. We used the traffic matrices provided by SNDLib for the different networks. For each demand, following [2], we then divide it into four *different categories of flows*: video streaming, web services, voice over IP, and online gaming. As shown in Table 1, each category has a different rate of traffic and corresponds to a different proportion of the total traffic given by [16]. Furthermore, each category has to pass through a *different chain of network functions* given in the table. As an example, the video traffic has a rate of 4Mbps, represents 69.9% of the total traffic and has to go through, first, a Network Address Translator, then, a Firewall, a Traffic Monitor, a Video Optimizer Controller and, last, an Intrusion Detection Prevention System.

For *evolution of the traffic during time*, we consider a *typical daily traffic* of France Telecom ISP link given in Figure 1. Traffic at night is about three times lower than during the day. It can be well approximated by considering only a small number of traffic levels. Indeed, an operator can achieve most of the possible energy savings by doing a small number of changes of its routing configurations during the day [17]. This is a great practical use as operators prefer to carry out as few as possible changes in order to minimize the chance of introducing errors or network instabilities. We thus consider 5 levels of traffic in the following.

Validation of the Heuristic Algorithm. We compare the results obtained by the heuristic algorithm, GreenChains, with the optimal results given by the integer linear program on a small network, `pdh`, with 11 nodes and 64 links. We consider instances with an increasing complexity:



(a) Execution time (log scale).

(b) Energy

Figure 2: Comparison of the performances of the heuristic algorithm, GreenChains, vs. the ILP for the `pdh` network.

the number of demands varies from 4 to 40. Note that we consider multiple of 4 demands, as the traffic between a pair of nodes is divided into four differents demands corresponds to different categories of traffic. We compare the execution times of the ILP model and of the algorithm in Figure 2a. The experiments are made on a Intel(R) Xeon(R) CPU E5620 @2.40GHz 16 cores with 24GB of RAM. We see that the ILP model can be used to solve the problem with a reasonable time for a maximum number of 16 demands. In this case, it takes around 23 minutes to return the optimal solution. The increase is exponential: for 20 demands, the execution time is almost 3 hours. GreenChains, on the contrary, is a lot faster. It returns a solution in less than 1 second (0.7 s) for 20 demands. The increase is linear. It solves an instance with 40 demands in 1.35 s and the all-to-all instance (with 440 demands) considered in the following in less than 8 s. We see that it is impractable to use the ILP for large numbers of demands, and thus we use the GreenChains for the experiments on larger networks in the following. The results in terms of energy are given in Figure 2b. GreenChains finds results within a precision between 0.2% to 22% for the different number of demands. We consider this as good results given the difficulty of the EE-SFCP problem. Moreover, it means that the potential energy savings of using dynamic traffic and virtualization are in fact even greater than the one presented in the following.

Energy Savings during the day. We compare three scenarios in the experiments:

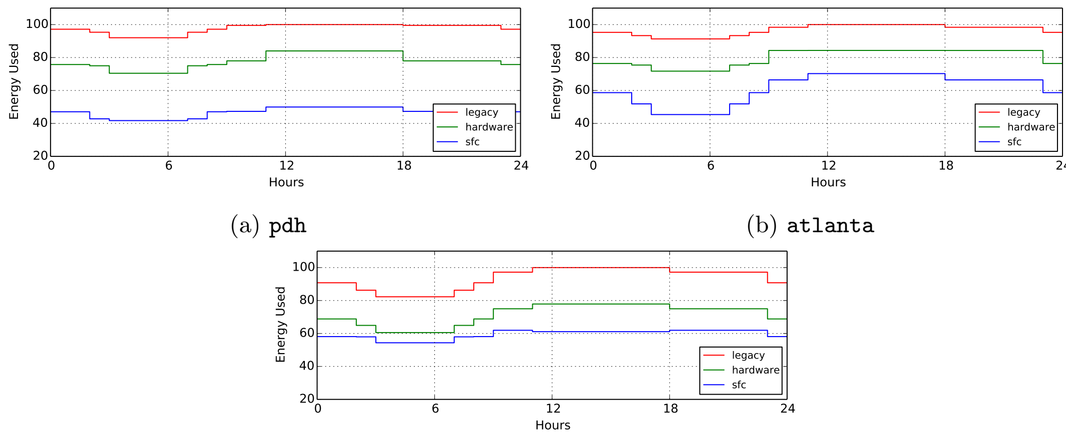
- *Legacy scenario.* This scenario corresponds to the one of a legacy network, whose operator does not try to reduce the energy consumption of its network. Its goal is to minimize the total bandwidth used while respecting the link capacity and the chain constraints. This scenario is used as a *baseline for comparison* for the energy-aware algorithms.

- *Hardware scenario.* The hardware scenario corresponds to the one of an SDN (non virtualized) networks in which an operator tries to reduce its energy consumption by adapting the routing to the demands. In this scenario, the network functions are carried out by some specific hardware placed at given positions in the network.

- *NFV scenario.* The NFV scenario is the one of a virtualized SDN network in which generic hardware nodes can execute any virtual network functions. This is the scenario solved by the ILP presented in Section 2

Results for the three networks are given in Figure 3. The baseline energy consumption of 100 corresponds to the maximum energy used in the legacy scenario.

First, we observe that, even if the goal in the *baseline scenario* is not to reduce the energy consumption but to minimize the total bandwidth usage, the energy used varies and is smaller at night for all networks. The explanation is that we used a mixed power model for which energy



(a) `pdh` (b) `atlanta`

(c) `germany`.

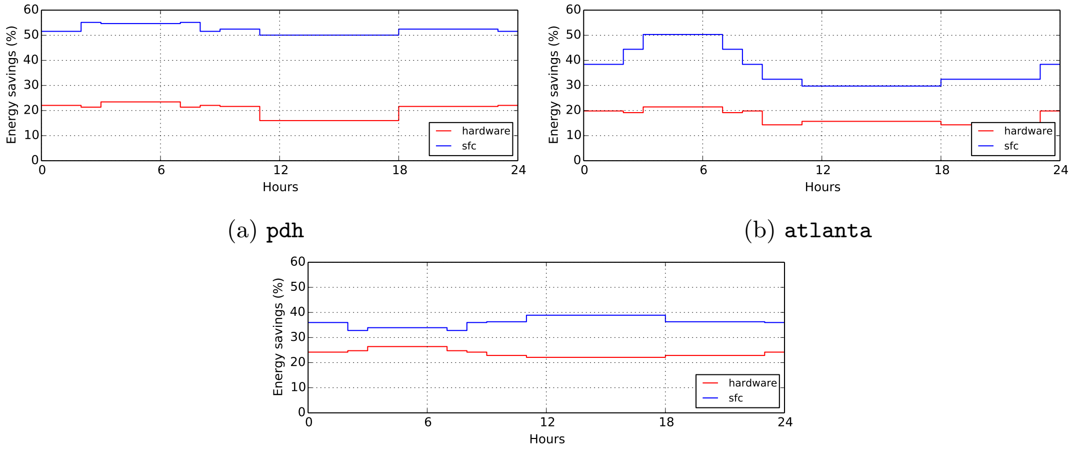
Figure 3: Energy used during the day for the *legacy, hardware and NFV scenarios* (The baseline 100 corresponds to the maximum energy used in the legacy scenario).

consumption depends for a part to the level of traffic which goes through the network links. Thus, during the night, even if no equipments (links or cores) is put into sleep mode, the energy used is smaller than during the day in which the traffic is higher. The variation is between 10 to almost 20% for the three networks considered.

We then focus on the energy which can be saved in the *hardware scenario* by trying to aggregate the traffic towards a small number of links and cores, while respecting the constraints given by the service chains and the capacities. The savings are between 15% and 27%, e.g. 22% at noon for `germany`. *The savings are here limited by the service chains: the flexibility for the choices of paths is limited in this scenario, as the network functions are executed by harware located at fixed positions in the network.*

The savings are a lot higher in the *nfv scenario*: between 30 to 55%. There are, for example, 50% at 6am and 30% at noon for `atlanta`, and always 50% or more for tt pdh. The explanation is that the *possibility to set VNF dynamically provides a larger flexibility to the algorithm to set the paths for the demands, leading to better aggregation, and thus energy savings.* We also observe a *surprising phenomena* in the results, which is best seen in Figure 4, when plotting the data of Figure 4c in terms of savings. The `germany` network is chosen as an example. The savings (compared to the legacy scenario) are almost constant during the day. They oscillate between 27% at 6 am to 22% at noon. The intuition would be that the savings are a lot larger during the night as the traffic is a lot lower, see Figure 1. Even more surprising, the savings in the nfv scenario for `germany` are lower during the night than during the day: 33% at 2am and 39% at noon. The explanation of this phenomenon is that, as we discuss it above, the energy consumption of the legacy scenario also decreases at night. The ratio harware/legacy or nfv/legacy scenario stays close to constant.

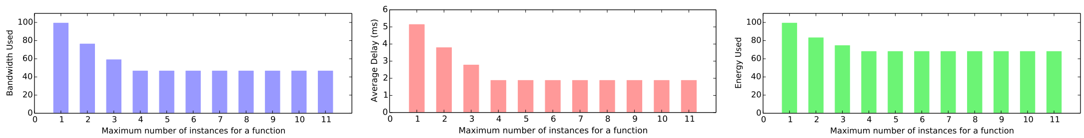
NFV occurrences. We look at the impact of the number of occurrences of the functions in the network on main metrics. We choose an instance, that we can solve using the ILP, the `pdh network` with 16 demands. This way, we can easily add constraints on the number of function occurrences in the model. We show in Figure 5 results for this instance about (a) bandwidth



(a) `pdh` (b) `atlanta`

(c) `germany`.

Figure 4: Energy savings (compared to the legacy scenario) during the day



(a) Bandwdith. (b) Average delay. (c) Energy Savings.

Figure 5: Impact of the maximum number of network function occurrences (bottom).

usage (a), (b) path delay, and (c) energy savings. We see that, when the number of nodes executing the same function is low, the bandwidth usage and the path delay is very high. As a matter of fact, the paths of the demands have to go through these nodes, which may be far away from a shortest path in the network. These results further explain the better efficiency of the NFV scenario over the hardware scenario. With virtualization, the cost to deploy several occurrences of a network function is low in comparison to the hardware scenario in which some specific hardware has to be installed. However, we see that the bandwdith and the average delay decrease quickly, when the number of occurrences of a function increases. This is an important result, as it shows that the energy savings using NFV can be obtained without installing a large number of NFV. In the case of `phd` which we take as example, we obtain almost optimal results in terms of bandwidth, delay, and energy savings with a maximum of 4 replicates per function.

5 Conclusion

In this work, we model the problem of minimizing the energy consumption of a network in which flows have to go through a chain of network functions. We propose an Integer Linear Program to solve the problem on small networks and an efficient heuristic algorithm to solve it on larger instances. We first show the impact of SDN technologies allowing dynamic routing on the energy consumption of the network. We obtain savings of around 20%. We then show that the energy savings may be largely increased by using network function virtualization. Indeed, being able to choose dynamically the position of the virtualized functions leads to further savings between 30 and 55% for different networks.

References

[1] D. Matsubara, T. Egawa, N. Nishinaga, V. P. Kafle, M.-K. Shin, and A. Galis, "Toward future networks: a viewpoint from itu-t," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 112–118, 2013.

[2] M. Savi, M. Tornatore, and G. Verticale, "Impact of processing costs on service chain placement in network functions virtualization," in *Network Function Virtualization and Software Defined Network (NFV-SDN), 2015 IEEE Conference on*, Nov 2015, pp. 191–197.

[3] A. Gupta, M. F. Habib, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Joint virtual network function placement and routing of traffic in operator networks," *UC Davis, Davis, CA, USA, Tech. Rep*, 2015.

[4] B. Martini, F. Paganelli, P. Cappanera, S. Turchi, and P. Castoldi, "Latency-aware composition of virtual functions in 5g," in *1st IEEE Conference on Network Softwarization (NetSoft)*, 2015, pp. 1–6.

[5] A. Mohammadkhan, S. Ghapani, G. Liu, W. Zhang, K. Ramakrishnan, and T. Wood, "Virtual function placement and traffic steering in flexible and dynamic software defined networks," in *The 21st IEEE Int. Workshop on Local and Metropolitan Area Networks*, 2015, pp. 1–6.

[6] R. Riggio, A. Bradai, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and T. Ahmed, "Virtual network functions orchestration in wireless networks," in *Network and Service Management (CNSM), 2015 11th International Conference on.* IEEE, 2015, pp. 108–116.

[7] L. Chiaraviglio, M. Mellia, and F. Neri, ""Minimizing ISP Network Energy Cost: Formulation and Solutions"," in *IEEE/ACM Trans. Netw.'11.*

[8] E. L. Rouzic, E. Bonetto, L. Chiaraviglio, F. Giroire, F. Idzikowski, F. Jimenez, C. Lange, J. Montalvo, F. Musumeci, I. Tahiri, A. Valenti, W. V. Heddeghem, Y. Ye, A. Bianco, and A. Pattavina, "Trend towards more energy-efficient optical networks," in *ONDM*, Brest, France, 2013, pp. 211–216.

[9] F. Giroire, J. Moulierac, and K. Phan, "Optimizing rule placement in software-defined networks for energy-aware routing." in *IEEE GLOBECOM)*, Austin, United States, December 2014.

[10] F. Giroire, J. Moulierac, T. K. Phan, and F. Roudaut, "Minimization of network power consumption with redundancy elimination," *Computer communications*, vol. 59, pp. 98–105, 2015.

[11] R. Bolla, C. Lombardo, R. Bruschi, and S. Mangialardi, "Dropv2: energy efficiency through network function virtualization," *IEEE Network*, vol. 28, no. 2, pp. 26–32, 2014.

[12] R. Mijumbi, "On the energy efficiency prospects of network function virtualization," *CoRR*, vol. abs/1512.00215, 2015. [Online]. Available: http://arxiv.org/abs/1512.00215

[13] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE.* IEEE, 2008.

[14] L. Niccolini, G. Iannaccone, S. Ratnasamy, J. Chandrashekar, and L. Rizzo, "Building a power-proportional software router," in *Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC 12)*, 2012, pp. 89–100.

[15] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[16] C. V. N. Index, "Cisco visual networking index: Forecast and methodology 2015-2020," *White paper, CISCO)*, 2015.

[17] J. Araujo, F. Giroire, J. Moulierac, Y. Liu, and M. Remigiusz, "Energy efficient content distribution," *Computer Journal*, pp. 1–18, 2016.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399